

X


<https://swayam.gov.in>

https://swayam.gov.in/nc_details/NPTEL

reviewer4@nptel.iitm.ac.in ▾

[NPTEL \(https://swayam.gov.in/explorer?ncCode=NPTEL\)](https://swayam.gov.in/explorer?ncCode=NPTEL) » **Programming, Data Structures And Algorithms**
Using Python (course)

Announcements (announcements)

About the Course (https://swayam.gov.in/nd1_noc19_cs40/preview) Ask a Question (forum)

Progress (student/home) Mentor (student/mentor)

Week 2 Programming Assignment

Due on 2019-08-22, 23:59 IST

Write three Python functions as specified below. Paste the text for all three functions together into the submission window. Your function will be called automatically with various inputs and should return values as specified. Do not write commands to read any input or print any output.

- You may define additional auxiliary functions as needed.
- In all cases you may assume that the value passed to the function is of the expected type, so your function does not have to check for malformed inputs.
- For each function, there are normally some public test cases and some (hidden) private test cases.
- "Compile and run" will evaluate your submission against the public test cases.
- "Submit" will evaluate your submission against the hidden private test cases. There are 10 private test cases, with equal weightage. You will get feedback about which private test cases pass or fail, though you cannot see the actual test cases.
- Ignore warnings about "Presentation errors".

1. Write a function `intreverse(n)` that takes as input a positive integer `n` and returns the integer obtained by reversing the digits in `n`.

Here are some examples of how your function should work.

```
>>> intreverse(783)
387
>>> intreverse(242789)
987242
>>> intreverse(3)
```

 Course
outline

 How to access
the portal

 Week 1:
Introduction

Week 1 Quiz

 Week 2: Basics
of Python

Week 2 Quiz

 Week 2
Programming
Assignment

 Week 2
Programming
Assignment
(/noc19_cs40/progassignment?
name=90)

 Week 3: Lists,
inductive
function

definitions, sorting

Week 3 Programming Assignment

Week 4: Sorting, Tuples, Dictionaries, Passing Functions, List Comprehension

Week 4 Quiz

Week 4 Programming Assignment

Week 5: Exception handling, input/output, file handling, string processing

Week 5 Programming Assignment

Week 6: Backtracking, scope, data structures; stacks, queues and heaps

Week 6 Quiz

Week 7: Classes, objects and user defined datatypes

Week 7 Quiz

Week 8: Dynamic programming, wrap-up

Week 8 Programming

```

def intreverse(s):
    3

```

- Write a function `matched(s)` that takes as input a string `s` and checks if the brackets "(" and ")" in `s` are matched: that is, every "(" has a matching ")" after it and every ")" has a matching "(" before it. Your function should ignore all other symbols that appear in `s`. Your function should return `True` if `s` has matched brackets and `False` if it does not.

Here are some examples to show how your function should work.

```

>>> matched("zb%78")
True
>>> matched("(7)(a)")
False
>>> matched("a)*(?)")
False
>>> matched("((jkl)78(A)&l(8(dd(FJI:),):)?)")
True

```

- Write a function `sumprimes(l)` that takes as input a list of integers `l` and returns the sum of all the prime numbers in `l`.

Here are some examples to show how your function should work.

```

>>> sumprimes([3,3,1,13])
19
>>> sumprimes([2,4,6,9,11])
13
>>> sumprimes([-3,1,6])
0

```

Sample Test Cases

| | Input | Output |
|-------------|--|--------------------|
| Test Case 1 | <code>intreverse(31511)</code> | 11513 |
| Test Case 2 | <code>intreverse(4)</code> | 4 |
| Test Case 3 | <code>intreverse(15135324234235)</code> | 5324324235 3151 |
| Test Case 4 | <code>matched("a3qw3;4w3(aasdgsd)((agadsgdsgag)agaga)")</code> | True |
| Test Case 5 | <code>matched("(ag(Gaga(agag)Gaga)GG)a)33)cc(")</code> | False |
| Test Case 6 | <code>matched("(adsgdsg(agaga)a)")</code> | False |
| Test Case 7 | <code>matched("15ababa.agaga[[[[")</code> | True |

Assignment**Download videos****Text Transcripts****Online Programming Test - Sample****Online Programming Test 1, 26 Sep 2019, 09:30-11:30****Online Programming Test 2, 26 Sep 2019, 20:00-22:00**

| | | |
|--------------|---|--------|
| Test Case 8 | sumprimes([101, 93, 97, 44]) | 198 |
| Test Case 9 | sumprimes([1001, 393, 743, 59]) | 802 |
| Test Case 10 | sumprimes([11, 11, 11, 13, 11, -11]) | 57 |
| Test Case 11 | intreverse(368) | 863 |
| Test Case 12 | intreverse(798798) | 897897 |
| Test Case 13 | intreverse(7) | 7 |
| Test Case 14 | matched("(7) (a)") | False |
| Test Case 15 | matched("a)*(?)") | False |
| Test Case 16 | matched("((jkl) 78 (A) &1 (8 (dd (FJ I:),) :) ?) ")") | True |
| Test Case 17 | sumprimes([17, 51, 29, 39]) | 46 |
| Test Case 18 | sumprimes([-3, -5, 3, 5]) | 8 |
| Test Case 19 | sumprimes([4, 6, 15, 27]) | 0 |

The due date for submitting this assignment has passed.

As per our records you have not submitted this assignment.

Sample solutions (Provided by instructor)

```

1 def intreverse(n):
2     ans = 0
3     while n > 0:
4         (d,n) = (n%10,n//10)
5         ans = 10*ans + d
6     return(ans)
7
8 def matched(s):
9     nested = 0
10    for i in range(0,len(s)):
11        if s[i] == "(":
12            nested = nested+1
13        elif s[i] == ")":
14            nested = nested-1
15            if nested < 0:
16                return(False)
17    return(nested == 0)
18
19 def factors(n):
20    factorlist = []
21    for i in range(1,n+1):
22        if n%i == 0:
23            factorlist = factorlist + [i]
24    return(factorlist)
25
26 def isprime(n):
27    return(factors(n) == [1,n])
28
29
30 def sumprimes(l):
31    sum = 0
32    for i in range(0,len(l)):
33        if isprime(l[i]):
34            sum = sum+l[i]

```

```
35     return(sum)
36
37
38 import ast
39
40 def tolist(inp):
41     inp = "["+inp+"]"
42     inp = ast.literal_eval(inp)
43     return (inp[0],inp[1])
44
45 def parse(inp):
46     inp = ast.literal_eval(inp)
47     return (inp)
48
49 fncall = input()
50 lparen = fncall.find("(")
51 rparen = fncall.rfind(")")
52 fname = fncall[:lparen]
53 farg = fncall[lparen+1:rparen]
54
55 if fname == "intreverse":
56     arg = parse(farg)
57     print(intreverse(arg))
58 elif fname == "matched":
59     arg = parse(farg)
60     print(matched(arg))
61 elif fname == "sumprimes":
62     arg = parse(farg)
63     print(sumprimes(arg))
64 else:
65     print("Function", fname, "unknown")
66
67
```